Art Pathways: Getting Graphics into NGPS Format

Attila Vass
Software Engineer
R & D
Sony Computer Entertainment America

The Next Generation PlayStation Developers Seminar – June 1999 Confidential

Blue

SONY

COMPUTER
ENTERTAINMENT

The Next Generation PlayStation Developers Seminar

Attila Vass

Software Engineer
R & D
Sony Computer Entertainment America

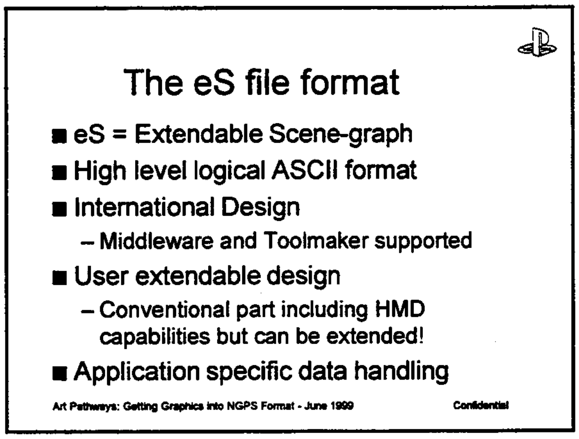
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

---

Overview

- The eS file format
- Getting the data from an Art-Package
- Optimizations

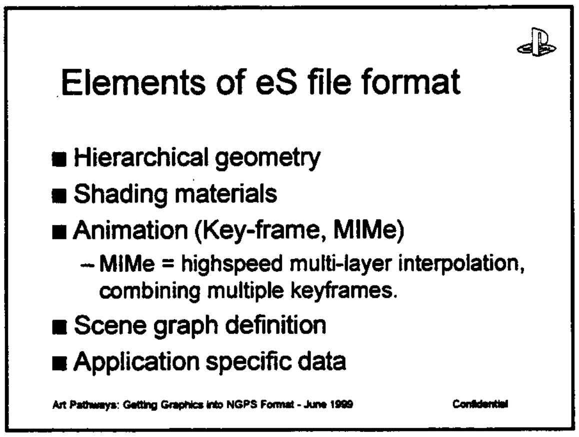
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



The eS file format

- eS = Extendable Scene-graph
- High level logical ASCII format
- International Design
  - Middleware and Toolmaker supported
- User extendable design
  - Conventional part including HMD capabilities but can be extended!
- Application specific data handling

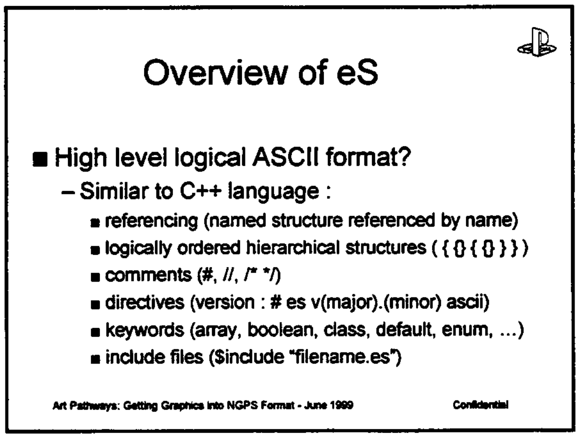
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Elements of eS file format

- Hierarchical geometry
- Shading materials
- Animation (Key-frame, MIMe)
  - MIMe = highspeed multi-layer interpolation, combining multiple keyframes.
- Scene graph definition
- Application specific data

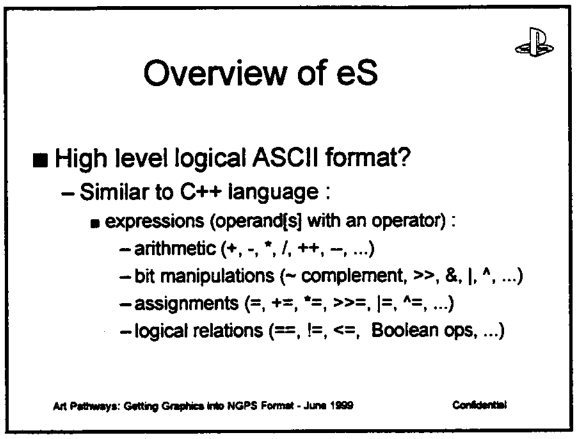
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Overview of eS

- High level logical ASCII format?
  - Similar to C++ language :
    - referencing (named structure referenced by name)
    - logically ordered hierarchical structures ({ {} { {} } })
    - comments (#, //, /* */)
    - directives (version : # es v(major).(minor) ascii)
    - keywords (array, boolean, class, default, enum, ...)
    - include files ($include "filename.es")

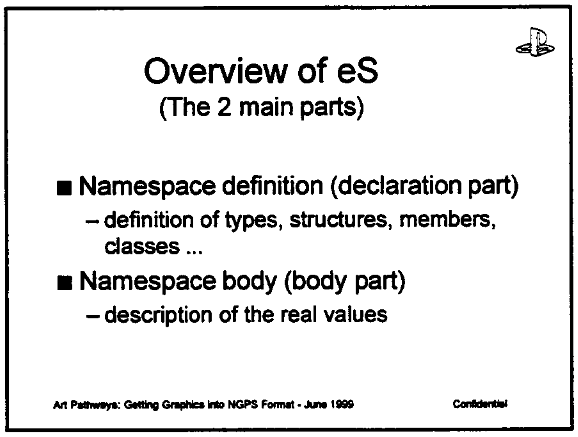
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Overview of eS

- High level logical ASCII format?
  - Similar to C++ language :
    - expressions (operand[s] with an operator) :
      - arithmetic (+, -, *, /, ++, –, ...)
      - bit manipulations (~ complement, >>, &, |, ^, ...)
      - assignments (=, +=, *=, >>=, |=, ^=, ...)
      - logical relations (==, !=, <=, Boolean ops, ...)

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

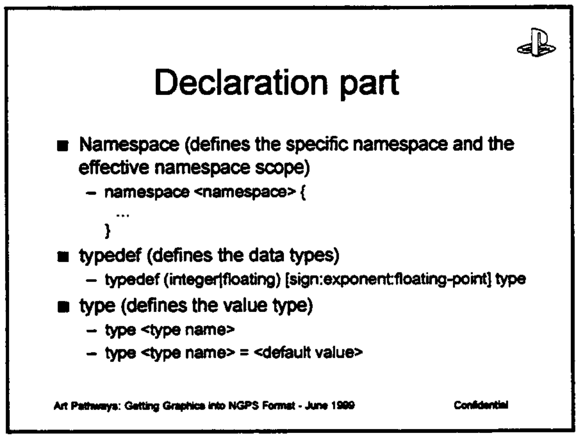


Overview of eS

(The 2 main parts)

- Namespace definition (declaration part)
  - definition of types, structures, members, classes ...
- Namespace body (body part)
  - description of the real values

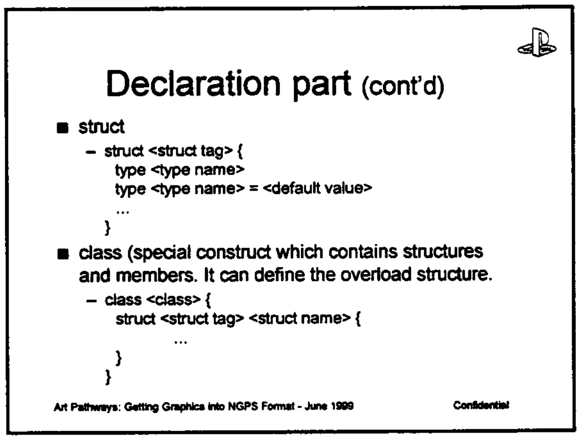
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Declaration part

- Namespace (defines the specific namespace and the effective namespace scope)
  - namespace <namespace> {
    ...
    }
- typedef (defines the data types)
  - typedef (integer|floating) [sign:exponent:floating-point] type
- type (defines the value type)
  - type <type name>
  - type <type name> = <default value>

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Declaration part (cont'd)

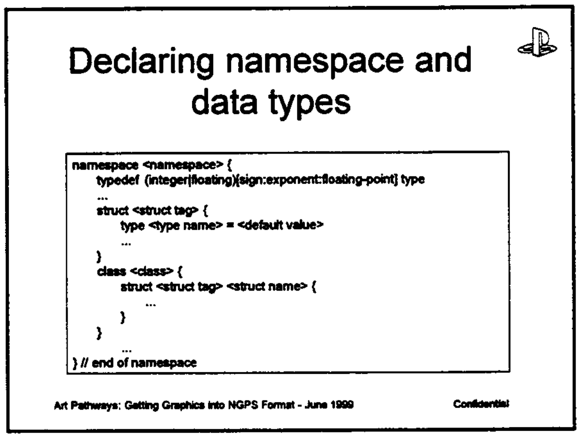
- struct
  - struct <struct tag> {

```
    type <type name>
    type <type name> = <default value>
    ...
}
```

- class (special construct which contains structures and members. It can define the overload structure.
  - class <class> {

```
    struct <struct tag> <struct name> {
        ...
    }
}
```

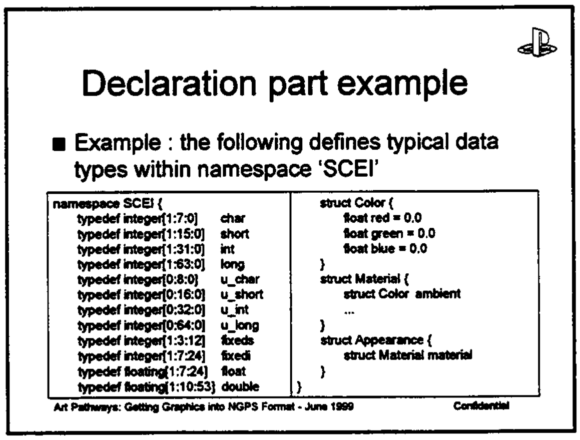
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Declaring namespace and data types

```
namespace <namespace> {
    typedef (integer|floating)[sign:exponent:floating-point] type
    ...
    struct <struct tag> {
        type <type name> = <default value>
        ...
    }
    class <class> {
        struct <struct tag> <struct name> {
            ...
        }
    }
        ...
} // end of namespace
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

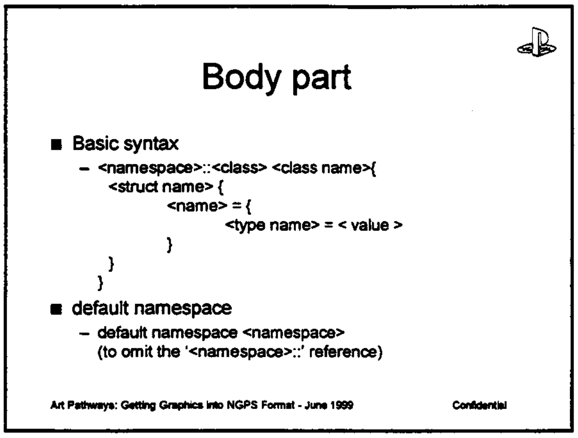


Declaration part example

- Example : the following defines typical data types within namespace 'SCEI'

```
namespace SCEI {
    typedef integer[1:7:0]      char
    typedef integer[1:15:0]     short
    typedef integer[1:31:0]     int
    typedef integer[1:63:0]     long
    typedef integer[0:8:0]      u_char
    typedef integer[0:16:0]     u_short
    typedef integer[0:32:0]     u_int
    typedef integer[0:64:0]     u_long
    typedef integer[1:3:12]     fixeds
    typedef integer[1:7:24]     fixedi
    typedef floating[1:7:24]    float
    typedef floating[1:10:53]   double
    struct Color {
        float red = 0.0
        float green = 0.0
        float blue = 0.0
    }
    struct Material {
        struct Color  ambient
        ...
    }
    struct Appearance {
        struct Material material
    }
}
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



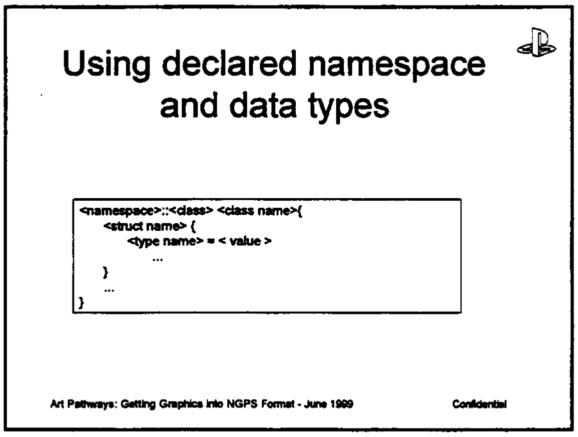
Body part

- Basic syntax
  - `<namespace>::<class> <class name>{`

```
<struct name> {
        <name> = {
                <type name> = < value >
        }
}
}
```

- default namespace
  - default namespace <namespace>
    (to omit the '<namespace>::' reference)

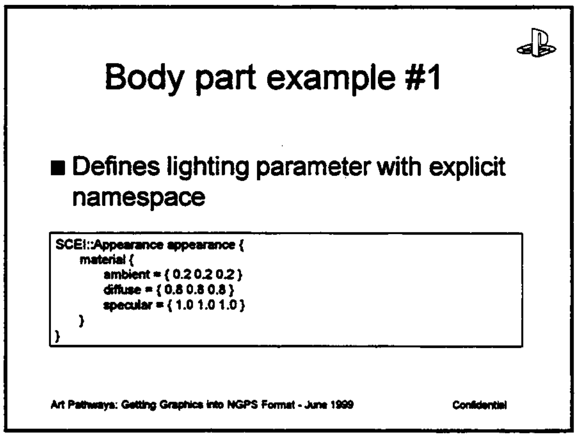
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Using declared namespace and data types

```
<namespace>::<class> <class name>{
    <struct name> {
        <type name> = < value >
            ...
    }
    ...
}
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

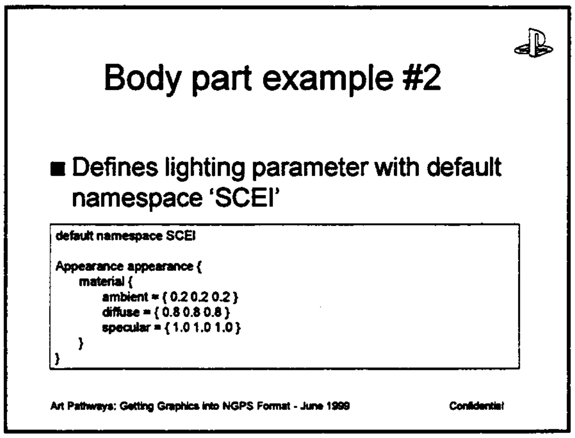


Body part example #1

- Defines lighting parameter with explicit namespace

```
SCEI::Appearance appearance {
    material {
        ambient = { 0.2 0.2 0.2 }
        diffuse = { 0.8 0.8 0.8 }
        specular = { 1.0 1.0 1.0 }
    }
}
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

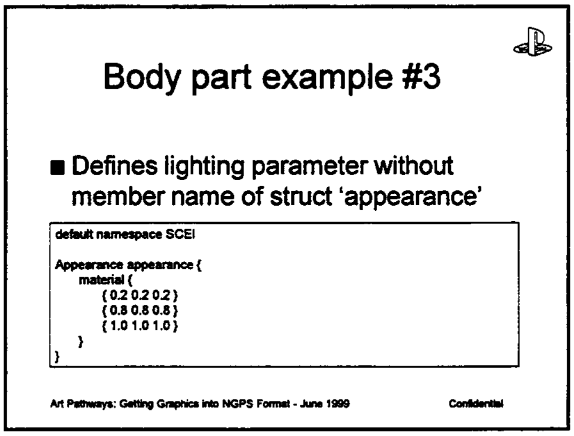


Body part example #2

- Defines lighting parameter with default namespace 'SCEI'

```
default namespace SCEI

Appearance appearance {
    material {
        ambient = { 0.2 0.2 0.2 }
        diffuse = { 0.8 0.8 0.8 }
        specular = { 1.0 1.0 1.0 }
    }
}
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

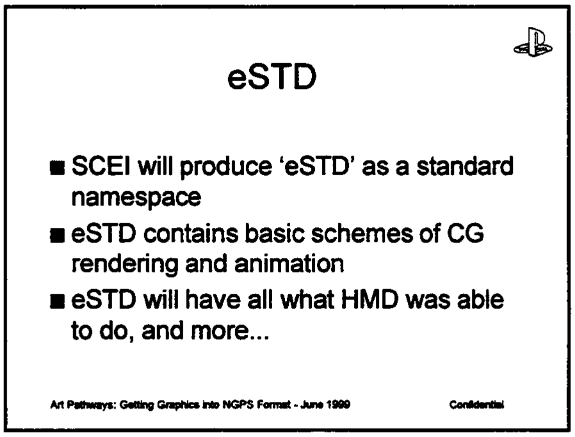


Body part example #3

- Defines lighting parameter without member name of struct 'appearance'

```
default namespace SCEI

Appearance appearance {
    material {
        { 0.2 0.2 0.2 }
        { 0.8 0.8 0.8 }
        { 1.0 1.0 1.0 }
    }
}
```

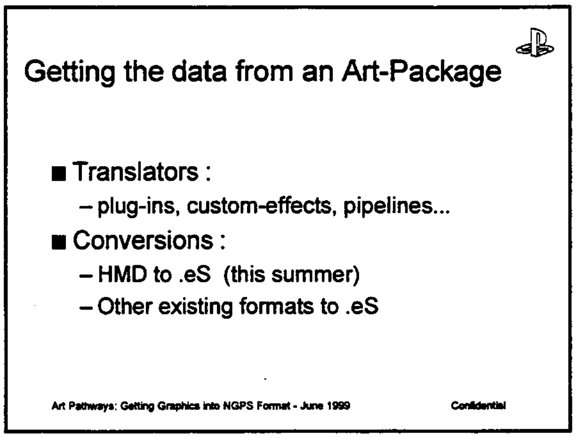
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



eSTD

- SCEI will produce 'eSTD' as a standard namespace
- eSTD contains basic schemes of CG rendering and animation
- eSTD will have all what HMD was able to do, and more...

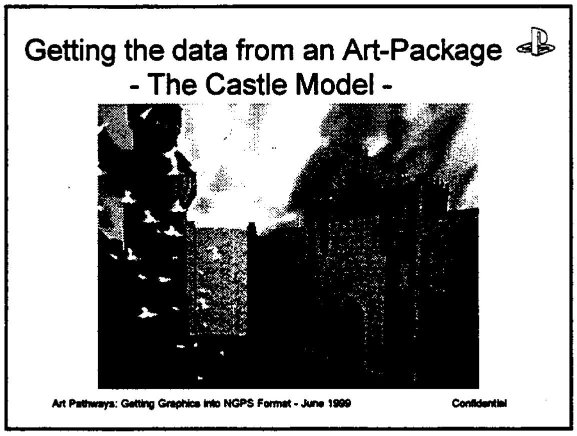
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package

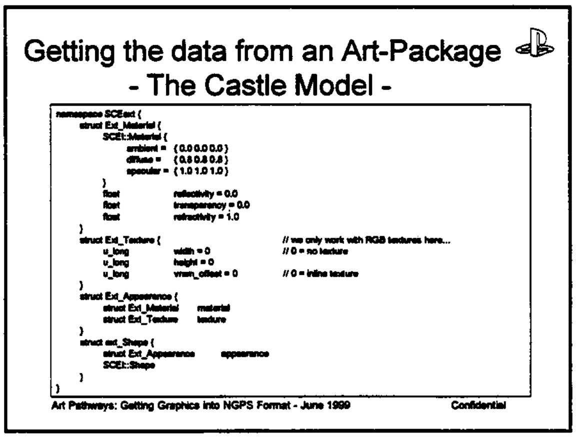
- Translators :
  - plug-ins, custom-effects, pipelines...
- Conversions :
  - HMD to .eS (this summer)
  - Other existing formats to .eS

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package
- The Castle Model -

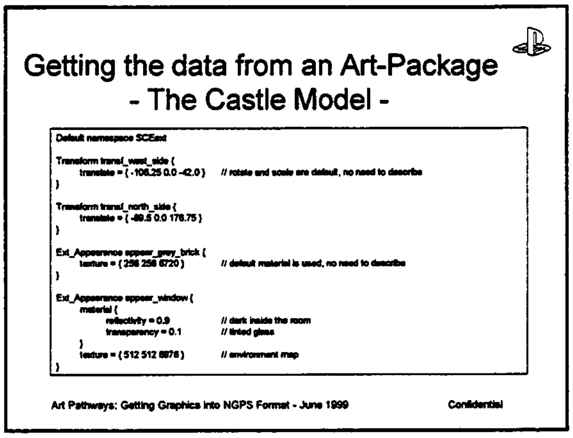
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package
- The Castle Model -

```
namespace SCEext {
	struct Ext_Material {
		SCEI::Material {
			ambient =  { 0.0 0.0 0.0 }
			diffuse =  { 0.8 0.8 0.8 }
			specular = { 1.0 1.0 1.0 }
		}
		float			reflectivity = 0.0
		float			transparency = 0.0
		float			refractivity = 1.0
	}
	struct Ext_Texture {					// we only work with RGB textures here...
		u_long		width = 0			// 0 = no texture
		u_long		height = 0
		u_long		vram_offset = 0		// 0 = inline texture
	}
	struct Ext_Appearence {
		struct Ext_Material	material
		struct Ext_Texture	texture
	}
	struct ext_Shape {
		struct Ext_Appearence		appearence
		SCEI::Shape
	}
}
```

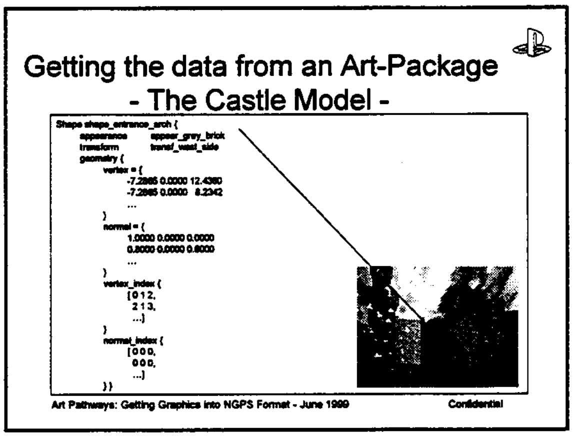
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package - The Castle Model -

```
Default namespace SCEext

Transform transf_west_side {
        translate = { -106.25 0.0 -42.0 }     // rotate and scale are default, no need to describe
}

Transform transf_north_side {
        translate = { -89.5 0.0 176.75 }
}

Ext_Appearance appear_grey_brick {
        texture = { 256 256 6720 }            // default material is used, no need to describe
}

Ext_Appearance appear_window {
        material {
                reflectivity = 0.9            // dark inside the room
                transparency = 0.1            // tinted glass
        }
        texture = { 512 512 6976 }            // environment map
}
```

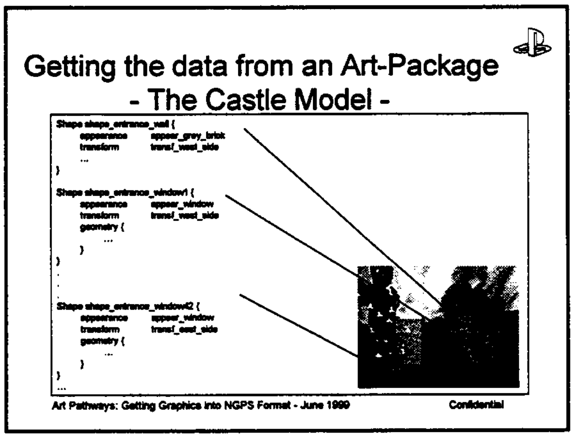
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package - The Castle Model -

```
Shape shape_entrance_arch {
        appearance      appear_grey_brick
        transform       transf_west_side
        geometry {
                vertex = {
                        -7.2865 0.0000 12.4380
                        -7.2865 0.0000  8.2342
                        ...
                }
                normal = {
                        1.0000 0.0000 0.0000
                        0.8000 0.0000 0.6000
                        ...
                }
                vertex_index {
                        [0 1 2,
                         2 1 3,
                         ...]
                }
                normal_index {
                        [0 0 0,
                         0 0 0,
                         ...]
                }}
```

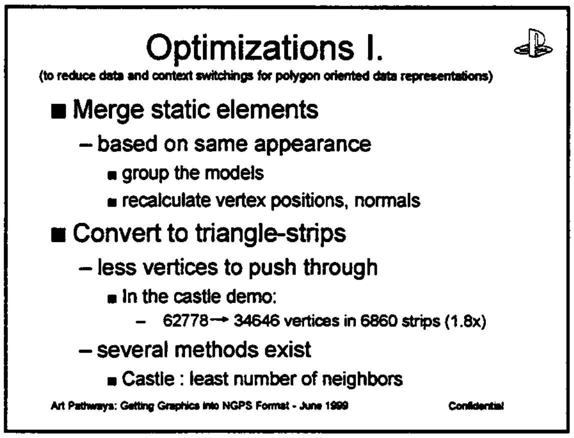
Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



Getting the data from an Art-Package - The Castle Model -

```
Shape shape_entrance_wall {
        appearance      appear_grey_brick
        transform       transf_west_side
        ...
}

Shape shape_entrance_window1 {
        appearance      appear_window
        transform       transf_west_side
        geometry {
                ...
        }
}
.
.
.
Shape shape_entrance_window2 {
        appearance      appear_window
        transform       transf_east_side
        geometry {
                ...
        }
}
...
```

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

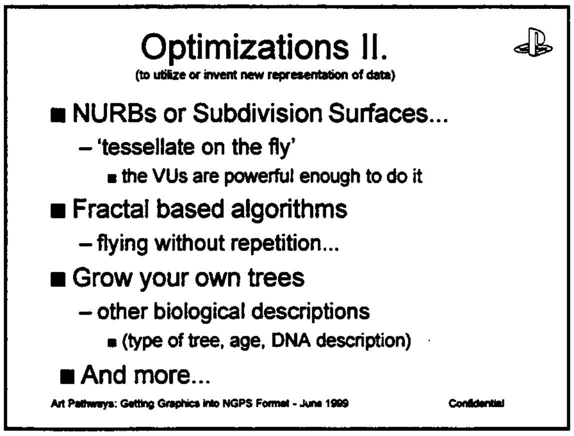


Optimizations I.

(to reduce data and context switchings for polygon oriented data representations)

- Merge static elements
  - based on same appearance
    - group the models
    - recalculate vertex positions, normals
- Convert to triangle-strips
  - less vertices to push through
    - In the castle demo:
      - 62778 → 34646 vertices in 6860 strips (1.8x)
  - several methods exist
    - Castle : least number of neighbors

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential

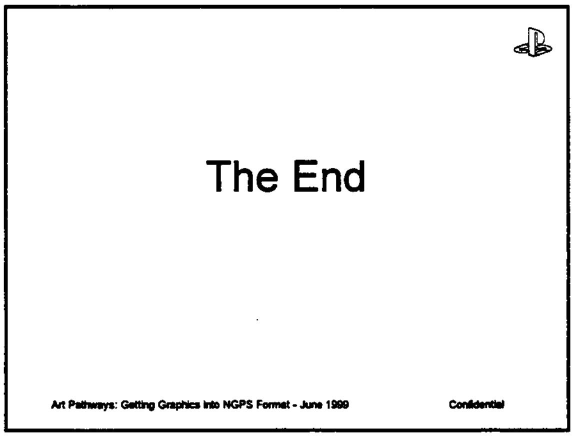


Optimizations II.

(to utilize or invent new representation of data)

- NURBs or Subdivision Surfaces...
  - 'tessellate on the fly'
    - the VUs are powerful enough to do it
- Fractal based algorithms
  - flying without repetition...
- Grow your own trees
  - other biological descriptions
    - (type of tree, age, DNA description)
- And more...

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential



The End

Art Pathways: Getting Graphics into NGPS Format - June 1999 Confidential